PRICE, PAMELA SUE. Simulation of a Communications System with Error Correction. (1974) Directed by: Dr. Richard Michael Willett. Pp. 43

It was the purpose of this study to introduce and illustrate all phases of communication and information theory. Basic theory presented includes a discussion of the uncertainty function, capacity, the binary symmetric channel, representations of codes, Shannon's Fundamental Theorem of Information Theory, group codes, coset partitioning, syndrome, and maximum likelihood decoding. These concepts are illustrated through the simulation of one particular communications system. Each segment of the system is simulated by a Fortran subroutine. These subroutines (listed in the appendices) were used to study the behavior of a noisy channel with and without error correction. Experiments were conducted to investigate the error correction capability of a special type of group code, the linear recursive group code. The procedures and results are summarized in the final chapter of this report. It was found that there was less error for the transmissions in which the coding procedures were employed and less error for one of the linear recursive group codes than for the other.

SIMULATION OF A COMMUNICATIONS SYSTEM WITH ERROR CORRECTION

by

Pamela Sue Price

A Thesis Submitted to
the Faculty of the Graduate School at
The University of North Carolina at Greensboro
in Partial Fulfillment
of the Requirements for the Degree
Master of Arts

Greensboro
August, 1974

Approved by

Michael Willett
Thesis Adviser

APPROVAL SHEET

This thesis has been approved by the following committee of the Faculty of the Graduate School at The University of North Carolina at Greensboro.

Thesis Adviser Michael Willett

Oral Examination Committee Members William A. Powell

K. A. Byrd

E E. Posey

12 July 1974
Date of Examination

ACKNOWLEDGMENT

The author would like to express her appreciation to Dr. Michael Willett for his time, patience, and guidance in the writing of this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Communication theory is concerned with the modeling and analysis of any communications system, a vehicle through which information is transmitted. There are different types of communication, such as speech, telephone conversations, radio, storage devices for computers, etc., but each has the following form:

(a) A source produces some message (the speaker, etc.).

(b) There is some device for transforming the information into an object which is acceptable to the channel.

(c) The channel is the medium through which the information is transmitted.

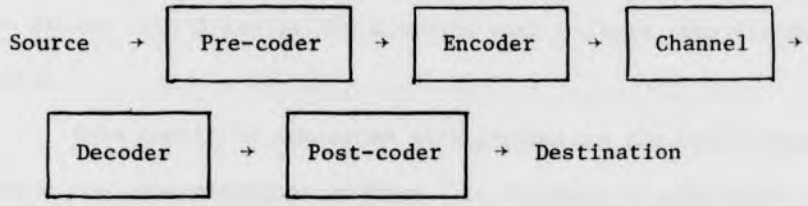
(d) There is something which acts on the output of the channel and makes a decision as to the identity of the original message.

(e) The message decided upon by (d) is delivered to its destination.

However, as the information passes through the channel, it may be altered by the effect of "noise", a term used for anything which causes errors in transmission. For example, there may be static in a radio transmission, or cross-talk or noise caused by lightning in telephone conversations. Thus, we are greatly concerned with finding reliable methods of altering the information, called coding, so that

errors produced in the channel can be corrected at their destination. A communications system which employs an error-correcting code differs from the original system in that it makes use of two more devices (one inserted just before and one just after the channel) which carry out the coding procedure. Figure 1.1 shows the model we will consider.

Figure 1.1 The Communications System



The source produces a message in English; the pre-coder converts the English to binary digits; the encoder alters the binary according to a coding procedure; this coded message passes through the channel where it is affected by noise; the decoder employs some decoding method in the attempt to extract the original message from the channel output; the post-coder converts this binary message into an English message which is sent to the destination. The accuracy of the final message depends on the capability of the coding/decoding procedure. We will use linear recursive group codes and a technique of decoding called maximum likelihood decoding.

Our particular pre-coder/post-coder scheme is as follows: The pre-coder receives the English message which is made up of the following thirty-two symbols:

ABCDEFGHIJKLMNOPQRSTUVWXYZ blank . , : ( )

Since there are thirty-two possible binary 5-tuples, there is a one to one correspondence between English symbols and 5-tuples. Thus, we associate a particular 5-tuple of zeros and/or ones with each English symbol. The pre-coder converts each symbol in the English message into a 5-tuple, thereby transforming the original message of length L into a string of zeros and/or ones of length 5L. After the decoding has been performed, the post-coder receives a binary message which it converts into English by reversing the above procedure. It segments the binary into 5-tuples and converts each 5-tuple into its English symbol.

This thesis is concerned with presenting the basic theory behind the communications problem. It attempts to illustrate the different phases of communication theory through the simulation of a particular communications system. Each segment of the system is simulated by a Fortran subroutine. Experiments are performed in an effort to explore the error correction capability of the codes used.

Since we are interested in the transmission of information, we would like some way of measuring the amount of information that can be conveyed through the communications system. In formulating a measure of information, we first need a measure of uncertainty. The reason for this can be seen from the following example. Suppose X is a random variable with four possible outcomes 1,2,3,4 each with equal probability of occurrence. If we try to guess the value of X, we have probability 1/4 of being correct. However, if we are told that the value of X is even and then asked to guess its value, we have a higher probability of being correct, namely a probability of 1/2.

In other words, there is less uncertainty about the value of X in the second situation than in the first situation. Thus, it appears that in order to arrive at a measure of information, we first need to formulate some measure of uncertainty. This is done in Chapter II.

Chapter III is a discussion of the channel in general - its transition matrix and the connection between the measures of uncertainty and information. The definition of capacity C is given and is then specialized to a particular channel, the binary symmetric channel, which is also described.

Chapter IV introduces the idea of a code, discusses error correction and error detection, and explains the method of decoding used in our experiments - maximum likelihood decoding. The rest of the chapter is devoted to the statement and explanation of the Fundamental Theorem of Information Theory.

Chapter V is a discussion of group codes. Several basic results concerning minimum weight, coset partitioning, syndrome, and maximum likelihood decoding are derived. A special type of group code, called the linear recursive group code, is then discussed.

Chapter VI is an explanation of the results of several experiments conducted using our particular communications system. The Fortran subroutines used in these experiments are listed and discussed in the appendices.

CHAPTER II

A MEASURE OF UNCERTAINTY

As pointed out in Chapter I, we may think of information received as uncertainty removed. Thus, in order to develop a way of measuring the information received upon observing the outcome of an event, we first need to develop an intuitive and quantitative measure of uncertainty associated with the possible occurrence of an event.

Let E be an event which occurs with probability p. Intuitively, the uncertainty associated with E is a function of the probability p alone and not of any other attribute of the event. Let $h(p)$ denote a numerical measure of uncertainty (as yet unspecified) associated with any event which occurs with probability p. Since p will be a number between 0 and 1, h will be defined on $(0,1]$, disregarding any event which has $p = 0$ probability of occurrence.

We now proceed to impose some very natural conditions on $h(p)$. An event which is certain to occur should have no uncertainty associated with it, so we require $h(1) = 0$. The more unlikely the occurrence of an event then the more uncertain we are about it, so $h(p)$ should be greater for smaller values of p. Thus, we require that $h(p)$ be monotonically decreasing on $(0,1]$. Small changes in p should produce small changes in $h(p)$, so h should be continuous. Furthermore, given two independent events E_1 and E_2 with associated

probabilities p_1 and p_2, respectively, the uncertainty associated with the joint event E_1E_2 should be the sum of the separate uncertainties, so we require that $h(p_1p_2) = h(p_1) + h(p_2)$. The following theorem shows that these conditions completely determine h.

2.1 Theorem: A function $h:(0,1] \to [0, +\infty)$ satisfies:

(1) $h(p)$ is continuous,

(2) $h(p)$ monotonically decreases to 0 as p approaches 1, and

(3) $h(pq) = h(p) + h(q)$

if and only if $h(p) = - c \log_2(p)$ for some constant $c > 0$.

Proof: (Necessity) Assume h is a function that satisfies the above conditions. Let $L(p) = \log_2(p)$. To show that $h(p) = -c \log_2(p)$ for some constant $c > 0$, let $g(p) \equiv - h(p)/L(p)$ on $(0,1)$ and define $g(1) = g(1/2)$. Thus, $h(p) = - g(p) L(p)$ on $(0,1)$. From (1) above, $h(pq) = h(p) + h(q)$, so $g(pq) L(pq) = g(p) L(p) + g(q) L(q)$. Since $L(p) = \log_2(p)$, we have $L(pq) = L(p) + L(q)$. Thus,

$$g(pq) [L(p) + L(q)] = g(p) L(p) + g(q) L(q)$$

$$L(p) g(pq) + L(q) g(pq) = L(p) g(p) + L(q) g(q)$$

$$L(p) [g(pq) - g(p)] = L(q) [g(q) - g(pq)] \qquad (2.2)$$

Let $q = p^x$, $p \in (0,1)$, $x > 0$. Equation (2.2) implies that $L(p)[g(p^{x+1}) - g(p)] = xL(p)[g(p^x) - g(p^{x+1})]$ so that $g(p) = g(p^{x+1}) - xg(p^x) + xg(p^{x+1})$ and

$$g(p) = (1 + x) g(p^{x+1}) - xg(p^x) \qquad (2.3)$$

Let $f(x) \equiv xg(p^x)$. Then equation (2.3) becomes $g(p) = f(x+1) - f(x)$

for which the most general solution is $f(x) = f(0) + xg(p)$. This conclusion requires continuity of h. Since $f(0) = 0g(p^0) = 0g(1) = 0$, we have $f(x) = xg(p)$. So, $xg(p^x) = xg(p)$. We see that

$$g(p^x) = g(p) \tag{2.4}$$

Pick b so that $0 < b < p < 1$ and let $x \equiv L(b)/L(p) > 0$. After some calculation, we find that $p^x = b$. So, from equation (2.4), $g(p^x) = g(b) = g(p)$. Therefore, $g(p)$ is constant on $(0,1)$. Since $h(p) > 0$, $L(p) < 0$, and $g(p) = - h(p)/L(p)$, $g(p)$ is a positive constant, $c = g(p)$.

(Sufficiency) Assume $h(p) = - c \log_2(p)$ for some constant $c > 0$. Condition (1) follows from the continuity of $\log_2(p)$ for all $p > 0$. Since $\log_2(p)$ is monotonically increasing, $h(p) = - c \log_2(p)$ is monotonically decreasing. Also, if $p = 1$, then $\log_2(p) = 0$. So $h(p) = 0$. Therefore, $h(p)$ monotonically decreases to 0 as p approaches 1, and condition (2) is satisfied.

$$h(pq) = - c \log_2(pq) = - c [\log_2(p) + \log_2(q)]$$
$$= - c \log_2(p) + [- c \log_2(q)] = h(p) + h(q)$$

so that condition (3) is satisfied. Q.E.D.

Note that if $c = 1$ and $p = 1/2$, then $h(p) = 1$. We will assume that an event with probability $p = 1/2$ of occurrence contains one unit of uncertainty, so that for the remainder of this section we will use $c = 1$. Thus, $h(p) = - \log_2(p)$.

Suppose X is an event with possible outcomes $X_1, X_2, \cdots, X_n$ and that the probability $P\{X = X_i\}$ is p_i. We define the uncertainty $H(X)$ associated with the event X as the average uncertainty of its possible outcomes. Since p_i is the

fraction of time we would expect $X = X_i$ to occur, we have

$$H(X) = \sum_{i=1}^{n} p_i \, h(p_i) = - \sum_{i=1}^{n} p_i \log_2(p_i) \, .$$

In Chapter III, we shall use the quantity $H(X)$ to define a measure of information.

CHAPTER III

BINARY SYMMETRIC CHANNEL

We have said that a channel is the medium through which coded information is transmitted. The objects which are input to the channel make up the input alphabet. The channel acts on this input and produces objects belonging to the output alphabet. We will be concerned only with those "discrete cases" where the alphabets are finite. Denote the input symbols by $x_1, x_2, \cdots, x_n$ and the output symbols by $y_1, y_2, \cdots, y_m$. Define $\rho_{ij} = P(y_j \mid x_i)$ to be the probability that y_j is received given that x_i was transmitted, and define the channel matrix to be the $n \times m$ matrix whose entries are the ρ_{ij}. We will assume that the channel is memoryless; i.e. each symbol is acted upon independent of previous transmissions. The output symbols do not depend upon previous inputs and outputs or upon the state of the channel; that is, the ρ_{ij} are constants independent of time. Thus, we will be concerned only with discrete memoryless channels.

Let X be the event representing the input of one symbol to the channel, and let Y be the associated event representing the output of one symbol. Let $p_i \equiv P\{X = x_i\}$ and $\rho_{ij} = P\{Y = y_j \mid X = x_i\}$, so that the joint distribution of the input X and the output Y is given by $P\{X = x_i, Y = y_j\} = p_i \rho_{ij}$. Thus, the distribution of Y is given by $P\{Y = y_j\} = \sum_{i=1}^{n} p_i \rho_{ij}$. Also, $P\{X = x_i, Y = y_j\} = P\{Y = y_j\} P\{X = x_i \mid Y = y_j\}$ so that

$$P\{X = x_i \mid Y = y_j\} = p_i \rho_{ij} \Big/ \sum_{k=1}^{n} p_k \rho_{kj} \equiv Q_{ij}.$$

We have previously defined

$$H(X) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

to be the uncertainty associated with the event X. Thus, the conditional uncertainty $H(X \mid Y)$ associated with the event X given that Y has occurred is intuitively defined as

$$H(X \mid Y) = -\sum_{i=1}^{n} \sum_{j=1}^{m} Q_{ij} \log_2(Q_{ij}).$$

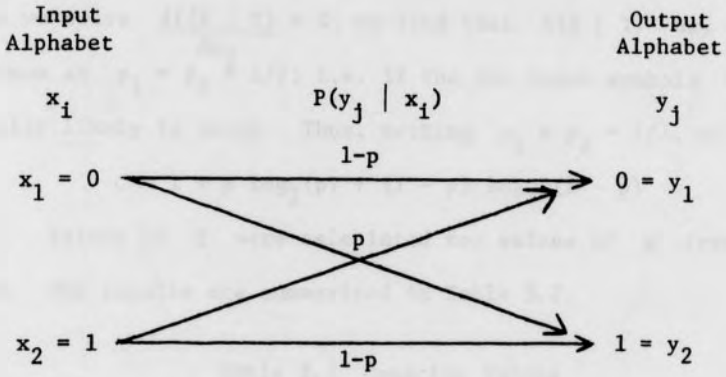
We can thus define the amount of information about one occurrence of the event X delivered to us by the channel to be

$$I(X \mid Y) \equiv H(X) - H(X \mid Y).$$

In words, the amount of information about X (or the amount of uncertainty about X which is removed by observing Y) is the uncertainty about X, $H(X)$, less the uncertainty $H(X \mid Y)$ about X left after observing Y. We see that, for a fixed channel, this amount is a function of the input distribution and of the fixed probabilities p_i. Therefore, $I(X \mid Y)$ would change if we were able to alter the input distribution. We might be able to "code" the original information so that the distribution is altered in such a way that $I(X \mid Y)$ is increased. For this reason, we define the capacity C of the channel to be the maximum average information over all possible input probabilities; i.e. $C \equiv \sup I(X \mid Y)$ where the supremum is taken over all input distributions $p_1, p_2, \cdots, p_n$ with $p_i > 0$ and $\Sigma p_i = 1$.

The channel model which we will investigate in detail is called the binary symmetric channel (BSC). It is shown in Figure 3.1.

Figure 3.1 Binary Symmetric Channel



where p is the probability of error per binary digit. The channel matrix for the BSC is

$$[\rho_{ij}] = [P(y_j \mid x_i)] = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}$$

since $\rho_{11} = P(y_1 \mid x_1) = 1 - p$

$\rho_{12} = P(y_2 \mid x_1) = p$

$\rho_{21} = P(y_1 \mid x_2) = p$

$\rho_{22} = P(y_2 \mid x_2) = 1 - p$

We will now compute the capacity of the BSC. Let $P\{X = x_1\} = P\{X = 0\} = p_1$ and $P\{X = x_2\} = P\{X = 1\} = p_2$ where $p_2 = 1 - p_1$ and $0 < p_1, p_2 \leq 1$.

Then

$$I(X \mid Y) = H(X) - H(X \mid Y)$$

$$= p \log_2(p) + (1 - p) \log_2 (1 - p)$$

$$- \{p_1(1 - p) + p_2 p\} \log_2 \{p_1(1 - p) + p_2 p\}$$

$$- \{p_1 p + p_2(1 - p)\} \log_2 \{p_1 p + p_2(1 - p)\}$$

When we solve $\frac{dI(X \mid Y)}{dp_2} = 0$, we find that $I(X \mid Y)$ has an absolute maximum at $p_1 = p_2 = 1/2$; i.e. if the two input symbols 0,1 are equally likely to occur. Thus, setting $p_1 = p_2 = 1/2$, we find that

$$C = 1 + p \log_2(p) + (1 - p) \log_2 (1 - p)$$

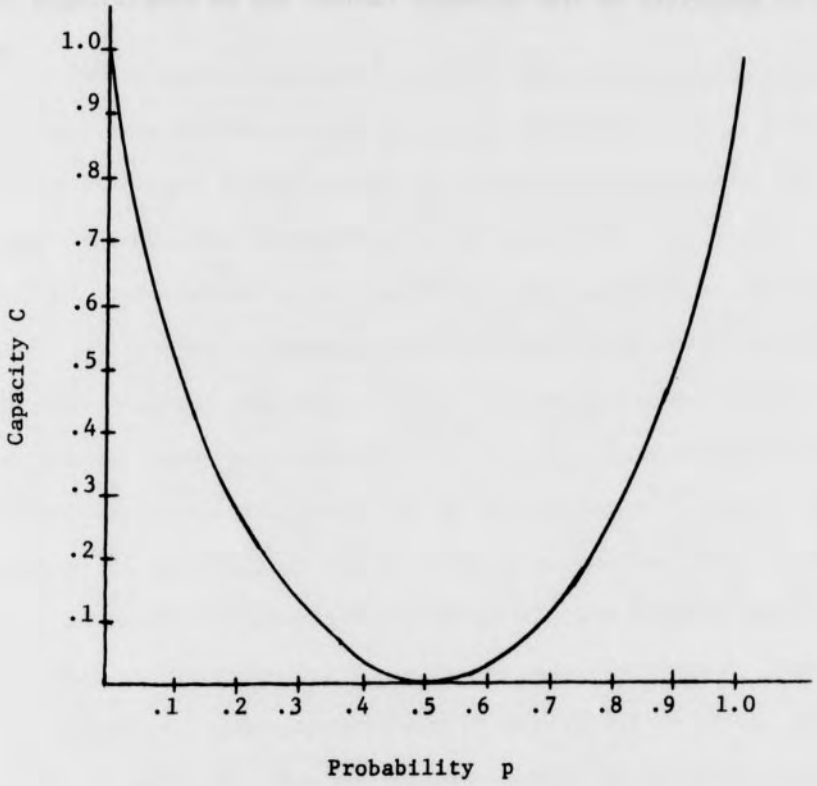
Values of C were calculated for values of p from 0.00 to 1.00. The results are summarized in Table 3.2.

Table 3.2 Capacity Values

p	C	p	C
.00	1.0000	.55	.0072
.05	.7136	.60	.0291
.10	.5310	.65	.0659
.15	.3902	.70	.1187
.20	.2781	.75	.1887
.25	.1887	.80	.2781
.30	.1187	.85	.3902
.35	.0659	.90	.5310
.40	.0291	.95	.7136
.45	.0072	1.00	1.0000
.50	.0000		

The above values of C are plotted in Figure 3.3.

Figure 3.3 Capacity



If an error has a 50% chance of occurring, we should expect to receive no information about X by observing Y, regardless of how X is coded. This is represented by $C = 0$ at $p = 1/2$. Note also that C is symmetric about $p = 1/2$. For values of $p > 1/2$, an error is more likely to occur than not. If we automatically converted each symbol received to the other symbol, then we have created a channel with $p' = 1/2 - (p - 1/2) = 1 - p$. For example, if $p = 1$, then

we know for certain that a symbol received indicates the other symbol was transmitted. We therefore receive as much information about X as for $p = 0$. This is an interpretation of the symmetry of C. The significance of the channel capacity will be explained in Chapter IV.

CHAPTER IV

SHANNON'S FUNDAMENTAL THEOREM

Since errors may occur in our attempt to transmit information through a particular system, we need to determine ways of altering the information, called coding, so that, upon transmission over a noisy channel, the information can be recovered. For example, suppose that the information to be transmitted is a sequence of zeros and/or ones. One method of coding would be to send four zeros for each zero and four ones for each one. Errors that occur due to noise in the channel may cause a particular 4-tuple to be converted into another. Our problem at the receiving end of the channel is to decide the identity of the original digit. This is called decoding. In this case, it would be reasonable to assume that the original digit is the one which occurs more frequently in the received 4-tuple. However, this method will not always result in the correct decision. Suppose we want to send 0. Then we transmit 0000. If no errors occur, 0000 will be received, and it will be correctly decoded as 0. If only one error occurs, the received 4-tuple will still contain more zeros than ones and it will be correctly decoded as 0. However, if two errors occur, the received 4-tuple will contain the same number of zeros as ones. Thus, we can make no decision, so we might ask for retransmission. If three or four errors occur, the received 4-tuple will contain more ones than zeros and it will be decoded incorrectly as 1. Thus, we see the limitations of this particular method of

coding and decoding. It can correct at most one error per four digits and detect at most two errors per four digits. Also, it takes four digits to represent one digit, and in one case we may need to ask for retransmission. We would like to find more efficient coding/decoding schemes. We will assume for the remainder of this discussion that the information to be transmitted is in the form of a string of zeros and/or ones.

We will now give a precise definition of a code. Let $F = \{0,1\}$ be the field with two elements and let $F_n = \{(u_1\ u_2\ \cdots\ u_n) \mid u_i \in F\}$ be the vector space of n-tuples over F under the usual addition and scalar multiplication. In particular, F_n is an abelian group under addition. An (n,k) code is a function $f:F_k \to F_n$. The image under the code of each k-tuple in F_k is some n-tuple in F_n. The set $C = \{f(u) \mid u \in F_k\}$ is called the set of codewords. We naturally require that f be one to one so that no two elements of F_k have the same image in F_n. For example, the function $f:F_2 \to F_5$ defined by the following is a $(5,2)$ code:

$$f(00) = (00000) \qquad f(10) = (10011)$$
$$f(01) = (01111) \qquad f(11) = (11100) \tag{4.1}$$

The set of codewords is $C = \{(00000), (01111), (10011), (11100)\}$. Note that f is one to one.

Define $w(v)$, called the Hamming weight of the vector v, to be the number of ones in v. The Hamming distance $d(u,v)$ between the two vectors u,v is the number of places in which u and v differ, so that $d(u,v) = w(u-v)$. For example, if $u = (10011)$, $v = (11110)$, then $w(u) = 3$, $w(v) = 4$, and $d(u,v) = w(u-v) = w(01101) = 3$. These

concepts can be used to define a particular coding/decoding scheme. The following sequence of operations is performed:

(a) A string of binary digits is segmented into k-tuples.

(b) Each k-tuple is converted into a n-tuple by f, and this string of n-tuples is transmitted.

(c) If $u \in F_n$ is transmitted and u' is received (u' might possibly contain errors), then a vector u'' in C which is "closest" to u' (in Hamming distance) is assumed to be the transmitted vector.

(d) The unique k-tuple mapped to u'' by f is assumed to be the original k-tuple.

The decoding scheme in (c) and (d) is called maximum likelihood decoding.

Let $d(C) \equiv \min \{d(u,v) \mid u, v \in C, u \neq v\}$ be the minimum distance between any two codewords. Let u be a transmitted vector and u' the received vector. Let $d(u,u') = e$. Thus, e is actually the number of errors introduced in u by the channel. If $0 \leq e \leq d(C) - 1$, then not enough errors occured for u' to be transformed into another codeword. Thus, we can at least detect this number of errors and ask for retransmission. If $0 \leq e \leq (d(C) - 1)/2$, then u' is still closest to u so that we would have successful error correction using maximum likelihood decoding. Thus, we see that the number $d(C)$ establishes certain upper bounds on the number of errors successfully detected and/or corrected using maximum likelihood decoding. One function of the coding theorist is to search for codes with large $d(C)$ without neglecting other important design considerations.

Other considerations in the search for "good" codes would be small error rates and high information rates. We would be interested in knowing if it is possible to find codes that combine these two; i.e. if there exist codes with small error rates which at the same time do not reduce appreciably the rate of transmission of information over the channel. A theorem discovered in 1948 by Shannon [4], properly called the Fundamental Theorem of Information Theory, is a statement to this effect. It states that it is possible to find codes which do the above, but it does not tell how to construct such codes. It is a non-constructive existence theorem and as such has provided nearly all the impetus to the search for the codes whose existence it guarantees. We will now introduce a few notions preliminary to the statement of the Fundamental Theorem.

Let $f:F_k \to F_n$ be a code. We might think of the fraction k/n as the rate of information when using the code. For example, if $k = 2$, $n = 5$, then each 5-tuple carries only two bits of information so that the information rate of the code is $2/5$. Assume that we use maximum likelihood decoding, and let $P(f)$ be the probability that an error goes undetected. For a large class of channels (including the BSC described in Chapter III) we have the following:

4.2 Theorem: (Shannon's Fundamental Theorem) Given a channel with capacity C, choose any positive numbers $R < C$ and $\epsilon > 0$. Then there exists a code $f:F_k \to F_n$ such that $R < (k/n) < C$ and $P(f) < \epsilon$.

In words, the theorem states that it is possible to encode the information so that the information delivery rate is arbitrarily close to the capacity of the channel while the probability of undetected error is arbitrarily small. This was proved by Shannon in an elegant but non-constructive manner. He showed that a code chosen at random would have the stated properties, on the average. More formal arguments have since been presented. It was not until 1972 [2] that someone first constructed a sequence of "good" codes with rates approaching channel capacity.

The Fundamental Theorem has a fairly strong converse, namely:

4.3. Theorem: Let $f_n : F_{k_n} \to F_n$ be any sequence of codes for which $C < M \leq (k_n/n)$, where C is the channel capacity and M is some constant. Then $\lim_{n\to\infty} P(f_n) = 1$.

In other words, if the rate of transmission of information for each code is greater than the channel capacity, we cannot hope for a reliable transmission. In fact, the probability that an error goes undetected approaches 1. This theorem, together with the Fundamental Theorem, clearly indicates that the theoretically motivated and derived notion of channel capacity actually has physical significance.

CHAPTER V

LINEAR RECURSIVE GROUP CODES

A code $f:F_k \to F_n$ for which the set C of codewords forms a vector subspace of the space of all binary n-tuples is called a linear group code. This is equivalent to requiring that C be a group under n-tuple addition since the only scalars are 0 and 1. Assume for the remainder of this section that C is a group. In the code defined by (4.1), we see that C is a group. Thus, the code is a linear group code.

We have previously defined $d(C)$ to be the $\min \{d(u,v) \mid u, v \in C, u \neq v\}$ or $\min \{w(u-v) \mid u, v \in C, u \neq v\}$. The group structure of C facilitates the calculation of $d(C)$ in the following way:

5.1 Lemma: If C is a group, then $d(C) = \min \{w(u) \mid u \in C, u \neq \theta\}$.

Proof: Since C is a group, then for each $v \in C$, $\{u - v \mid u \in C\} = C$. Therefore,

$$d(C) = \min \{w(u-v) \mid u - v \neq \theta, u, v \in C\}$$
$$= \min \{w(u) \mid u \neq \theta, u \in C\}.$$

Q.E.D.

Thus, we can find the minimum distance $d(C)$ between any two distinct codewords by calculating the minimum weight of all nonzero codewords. For the code (4.1), $d(C) = \min \{4,3,3\} = 3$.

We now proceed to describe an efficient representation for the code function f. Since f is one to one, we know that C is a k-dimensional subspace of F_n. Define a generator matrix for C to be any k by n matrix G' over F whose rows form a basis for C. If we put G' in reduced-echelon form, the resulting matrix G is of the form $[I_k \mid P]$ where P is k by $(n - k)$. Since row-reducing a matrix G' preserves the row space, we know that G is also a generator matrix for C. In the code (4.1), note that $\{(10011), (11100)\}$ is a basis for C. So, $G' = \left[\begin{array}{c|c} 10 & 011 \\ 11 & 100 \end{array}\right]$ is a generator matrix for C. By putting G' in reduced-echelon form, we obtain another generator matrix G for C: $G = \left[\begin{array}{c|c} 10 & 011 \\ 01 & 111 \end{array}\right]$.

The representation of f is as follows: Let $(x_1\ x_2\ \cdots\ x_k) \in F_k$ where the x_i are coordinates of a vector in the basis consisting of the rows of G. Therefore, $f(x_1\ x_2\ \cdots\ x_k) = (x_1\ x_2\ \cdots\ x_k)\ G$. Note that the original k-tuple is always the first k terms of the codeword. Such a representation is called a systematic code. The first k terms of the codeword are called the information symbols while the final $n - k$ terms are called the parity check symbols. In our example above,

$$f(00) = (00)\ G = (00) \begin{bmatrix} 10011 \\ 01111 \end{bmatrix} = (00000)$$

$$f(01) = (01)\ G = (01) \begin{bmatrix} 10011 \\ 01111 \end{bmatrix} = (01111)$$

$$f(10) = (10)\ G = (10) \begin{bmatrix} 10011 \\ 01111 \end{bmatrix} = (10011)$$

$$f(11) = (11)\ G = (11) \begin{bmatrix} 10011 \\ 01111 \end{bmatrix} = (11100)$$

which check with the original assignments in (4.1). Therefore, code (4.1) is a systematic code.

Results from basic vector space theory require that if C is a k-dimensional subspace of F_n, then there exists an $(n-k)$-dimensional subspace C' of F_n such that $C' = \{u \in F_n \mid v\,u^T = 0 \text{ for all } v \in C\}$. We define a parity check matrix for C to be any $(n-k)$ by n matrix H' over F whose rows form a basis for C'. We have the following:

5.2 Lemma: If $G = [I_k \mid P]$ is a generator matrix for C, then $H = [-P^T \mid I_{n-k}]$ is a parity check matrix for C.

Proof: Let $v = (x_1\ x_2\ \cdots\ x_n) \in C$. Then $v = (x_1\ x_2\ \cdots\ x_k)\,G$ so that $(x_{k+1}\ \cdots\ x_n) = (x_1\ \cdots\ x_k)\,P$. Form the $(n-k)$-tuple $vH^T = (x_1\ x_2\ \cdots\ x_k)\,(-P) + (x_{k+1}\ \cdots\ x_n)\,I_{n-k}$. Thus, $vH^T = (00\cdots0)$ if and only if $(x_{k+1}\ \cdots\ x_n) = (x_1\ x_2\ \cdots\ x_k)\,P$. This is satisfied for each $v \in C$. Thus, $vu^T = 0$ for each row u of H, and consequently for each row space of H. Thus, the rows of $H = [-P^T \mid I_{n-k}]$ form a basis for C'. H is therefore a parity check matrix. Q.E.D.

We previously determined that $G = \left[\begin{array}{cc|ccc} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{array}\right]$ and $P = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ for code (4.1). Thus,

$$H = [-P^T \mid I_3] = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

is a parity check matrix for C.

For each vector $u \in F_n$, the $(n-k)$-tuple uH^T is called the syndrome of u. We can now characterize the codewords C as the set of n-tuples whose syndrome is 0. We illustrate this with the above example. Note that $u = (11100) \in F_5$ is a codeword and

$$uH^T = (11100)\begin{bmatrix}011\\111\\100\\010\\001\end{bmatrix} = (000) = \theta$$

but $v = (11111) \in F_5$ is not a codeword and

$$vH^T = (11111)\begin{bmatrix}011\\111\\100\\010\\001\end{bmatrix} = (011) \neq \theta$$

Consider the additive cosets of C, $u + C = \{u + v \mid v \in C\}$. We know that the distinct cosets partition F_n. Using this fact, we can arrive at an algorithm which implements maximum likelihood decoding. Since any vector e in a particular coset of C can be used to represent that coset, we have that $e + C = e' + C$ for each e, e' in the same coset.

5.3 Theorem: Let e be chosen to have minimum weight over all other vectors in $e + C$. If $u' = e + u$ for some $u \in C$, then $d(u', u) \leq d(u', v)$ for all $v \in C$.

Proof: $d(u', u) = d(e + u, u) = d(e, \theta) = w(e) \leq w(e + (u + v))$

$\leq d(v + (u + e), \theta) = d(v + u', \theta)$

$= d(u', v)$. Q.E.D.

This fact represents an algorithm for decoding. Form an array, called the standard array, whose rows consist of the cosets of C, with $C = \theta + C$ the first row and θ the first vector in that row. Choose a representative e_i of minimum weight from each coset. The vector e_i is called the coset leader. This array is formed in Figure 5.4.

Figure 5.4 Standard Array (in general)

$$
\begin{array}{lllll}
\theta + C = \theta & u_1 & u_2 & \cdots & u_L \\
e_1 + C = e_1 & e_1 + u_1 & e_1 + u_2 & \cdots & e_1 + u_L \\
e_2 + C = e_2 & e_2 + u_1 & e_2 + u_2 & \cdots & e_2 + u_L \\
\vdots \quad \vdots & \vdots & \vdots & \vdots & \vdots \\
e_M + C = e_M & e_M + u_1 & e_M + u_2 & \cdots & e_M + u_L
\end{array}
$$

Every possible n-tuple will appear exactly once in the array. A received word u' is decoded into the codeword u which appears at the top of the column which contains u'. The preceeding theorem guarantees that $u \in C$ is a closest vector to u'. Figure 5.5 is the standard array for our previous example.

Figure 5.5 Standard Array (for code (4.1))

(00000) + C =	00000	11100	10011	01111
(10000) + C =	10000	01100	00011	11111
(01000) + C =	01000	10100	11011	00111
(00100) + C =	00100	11000	10111	01011
(00010) + C =	00010	11110	10001	01101
(00001) + C =	00001	11101	10010	01110
(00101) + C =	00101	11001	10110	01010
(01001) + C =	01001	10101	11010	00111

Assume $u' = (10010)$ is the received 5-tuple. Then $u' = e + u = (00001) + (10011)$, so u' is decoded as $u = (10011)$, the codeword which appears at the top of the column containing u'.

This decoding scheme can be further simplified through the use of 5.6 Lemma.

5.6 Lemma: Two vectors u, v are in the same coset of C if and only if they have the same syndrome.

Proof: $uH^T = vH^T$ if and only if (iff) $(u - v)\,H^T = 0$ iff $(u - v) \in C$ iff $(u - v) + C = C$ iff $u + C = v + C$. Q.E.D.

Since there are $2^n/2^k = 2^{n-k}$ cosets for C and 2^{n-k} different possible $(n-k)$-tuple syndromes, the above lemma establishes a one to one correspondence between syndromes and $(n-k)$-tuples. Thus, instead of storing the entire standard array, we need only store the coset leader and syndrome for each coset. To decode a received n-tuple u', we first calculate its syndrome, $s = u'H^T$. Next, we locate the coset leader e which has the same syndrome. Then we assume that $u' - e$ is the transmitted vector. Returning to our previous example, we see that there are eight cosets for C. We calculate the syndrome for each coset leader and store them together. They are given in Table 5.7.

Suppose $u' = (10010)$ is received. Then $s = u'H^T = (001)$. The coset leader $e = (00001)$ has the same syndrome. Thus, u' is decoded into $u = u' - e = (10010) - (00001) = (10011)$. Note that this result agrees with the preceeding technique for maximum likelihood decoding.

Table 5.7 Coset Leader/Syndrome Pairs

u	uH^T
(00000)	(000)
(10000)	(011)
(01000)	(111)
(00100)	(100)
(00010)	(010)
(00001)	(001)
(00101)	(101)
(01001)	(110)

We will now discuss a special type of group code, the linear recursive group code. Consider the linear recurrence relation

$$u_{n+k} = \alpha_1 u_{n+k-1} + \alpha_2 u_{n+k-2} + \cdots + \alpha_k u_n \qquad (5.8)$$

where $n = 0, 1, 2, \cdots, \alpha_i \in F = \{0, 1\}$, and addition is modulo 2. We call $f(x) = x^k - \alpha_1 x^{k-1} - \alpha_2 x^{k-2} - \cdots - \alpha_k$ the characteristic polynomial of the recursion. A value of u_k can be found given $u_0, u_1, \cdots, u_{k-1}$. Then u_{k+1} can be found from $u_1, u_2, \cdots, u_k$, etc. until u_{n+k} is found for any value of n. Note that a solution of (5.8) is a sequence of elements from F and that any linear combination of solutions is also a solution. Thus, the set of solutions forms a vector space (or a group under +). The space has dimension k, and a basis for this space would be the k different solutions in which one of the symbols $u_0, u_1, \cdots, u_{k-1}$ is 1 and and the rest are 0. Since this set of solutions to (5.8) forms a

group, and the n-tuples are derived from a linear recursion, the solutions (truncated after n positions) form an (n, k) group code, called a linear recursive group code.

The linear recursive group code is the type code used in our experiments in Chapter VI with a communications system. One such code is a (7,3) code defined by the linear recursion $u_{n+3} = u_{n+2} + u_n$ with characteristic polynomial $f(x) = x^3 + x^2 + 1$ (or $f(x) = x^3 - x^2 - 1$) and minimum weight 4. The other code considered is a (10,5) code defined by the recursion $u_{n+5} = u_{n+3} + u_{n+1} + u_n$ with characteristic polynomial $f(x) = x^5 + x^3 + x + 1$ (or $f(x) = x^5 - x^3 - x - 1$) and minimum weight 3.

The following is an example of how the encoding is performed using the above (7,3) linear recursive group code. If we start with the 3-tuple $(u_0, u_1, u_2) = (110)$, we find that $u_3 = 1$, $u_4 = 0$, $u_5 = 0$, $u_6 = 1$ so that $f(110) = (1101001)$.

Linear recursive codes have various alternate characterizations which simplify the analysis and implementation of these codes. However, most classes of linear recursive codes have been shown to be asymptotically bad in the sense that the ratio of minimum distance to codeword length approaches 0 for large codeword length.

The encoder, decoder, and syndrome subroutines in the appendices are based on using a linear recursive group code.

CHAPTER VI

EXPERIMENTS

In this section we wish to discuss several experiments which were performed with a particular communications system. The programs that were used in these experiments are documented in the appendices. Each segment of the communications system is in subroutine form so that it may be included or excluded for a particular experiment.

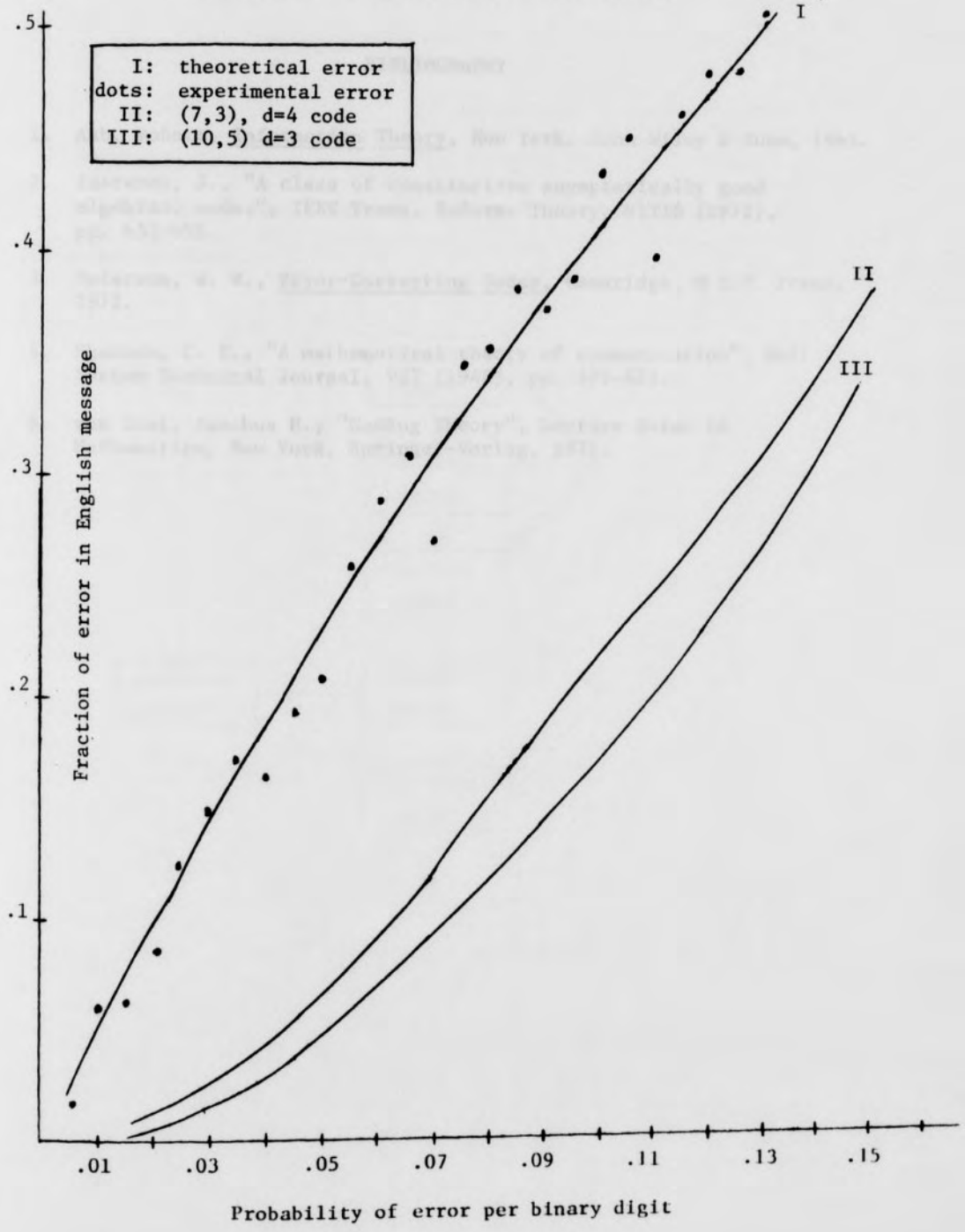
Suppose an English message is transmitted through the noisy communications system without error correction. Let us calculate the theoretical probability of error per English symbol. Recall that each English symbol is represented by a binary 5-tuple. Thus, the theoretical probability of error in one English symbol is the same as the probability of error in at least one of the five binary digits. This probability is 1 minus the probability of no error in each of the five binary digits, i.e. $1 - (1 - p)^5$ where p is the probability of error per binary digit for the binary symmetric channel. This function is plotted on Figure 6.1 as curve (I). The numerical values are given in Appendix A. For values of p from .005 to .150, the following 494 symbol message was pre-coded, transmitted, and post-coded:

> IN THIS PROGRAM THE FOLLOWING SEGMENTS OF A COMMUNICATIONS SYSTEM ARE EACH SIMULATED BY A FORTRAN SUBROUTINE: (ONE) PRECODER: CONVERTS ENGLISH TO BINARY. (TWO) ENCODER: CONVERTS BINARY TO CODEWORDS IN A CYCLIC ERROR CORRECTING CODE. (THREE) BINARY SYMMETRIC CHANNEL. (FOUR) DECODER: MAXIMUM LIKELIHOOD DECODING IS USED TO CONVERT THE RECEIVED WORDS TO THE NEAREST CODEWORDS. (FIVE) POSTCODER: CONVERTS BINARY TO ENGLISH. THE ENGLISH IS MADE UP OF THESE SYMBOLS: ABCDEFGHIJKLMNOPQRSTUVWX YZ .,:()

The fraction of error in the received English text was calculated for each value of p and is plotted on Figure 6.1 as dots. Notice the close agreement with the theoretical probability of error. The numerical values are given in Appendix A.

Two other experiments were performed in an effort to reduce the actual fraction of error, thereby insuring greater accuracy in transmission. The above message was re-transmitted for each value of p above, but now the encoder and decoder subroutines for a particular (n, k) linear recursive group code are added. In experiment (2), we used the (7,3) code referred to in Chapter V defined by $u_{n+3} = u_{n+2} + u_n$. In experiment (3), we used the (10,5) code also referred to in Chapter V defined by $u_{n+5} = u_{n+3} + u_{n+1} + u_n$. The experimental results are summarized in Appendix A and plotted on Figure 6.1. Note that the graph shows less error per English symbol for the transmissions in which the coding procedures were employed. Note also that the error is less for the (10,5) code than for the (7,3) code. The prediction of this relative error reduction is a significant but difficult area of coding theory.

Figure 6.1 Graph of Experimental Values



BIBLIOGRAPHY

1. Ash, Robert, Information Theory, New York, John Wiley & Sons, 1965.

2. Justesen, J., "A class of constructive asymptotically good algebraic codes", IEEE Trans. Inform. Theory, VIT18 (1972), pp. 652-656.

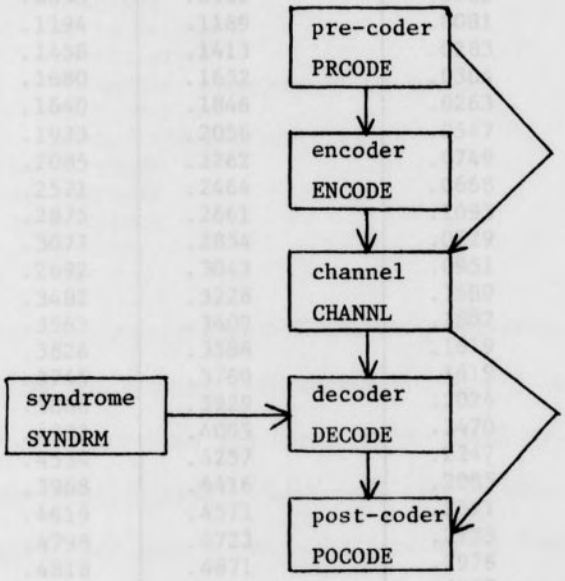
3. Peterson, W. W., Error-Correcting Codes, Cambridge, M.I.T. Press, 1972.

4. Shannon, C. E., "A mathematical theory of communication", Bell System Technical Journal, V27 (1948), pp. 379-423.

5. van Lint, Jacobus H., "Coding Theory", Lecture Notes in Mathematics, New York, Springer-Verlag, 1971.

APPENDICES

The subroutines in these appendices have the following inter-connection:



APPENDIX A

TABLE OF EXPERIMENTAL VALUES

	Experiment 1 (without error correction)		Experiment 2 (with error correction)	Experiment 3 (with error correction)
P	Actual error	Theoretical error	(7,3) linear recursive group code	(10,5) linear recursive group code
.005	.0162	.0248	.0020	.0000
.010	.0587	.0490	.0000	.0000
.015	.0628	.0728	.0142	.0081
.020	.0850	.0961	.0061	.0061
.025	.1194	.1189	.0081	.0081
.030	.1458	.1413	.0283	.0162
.035	.1680	.1632	.0304	.0202
.040	.1640	.1846	.0263	.0304
.045	.1923	.2056	.0547	.0425
.050	.2085	.2262	.0749	.0628
.055	.2571	.2464	.0668	.0466
.060	.2875	.2661	.1093	.0729
.065	.3077	.2854	.0729	.0830
.070	.2692	.3043	.0951	.0931
.075	.3482	.3228	.1539	.0972
.080	.3563	.3409	.1802	.1255
.085	.3826	.3586	.1619	.1296
.090	.3745	.3760	.1619	.1498
.095	.3866	.3929	.2024	.1660
.100	.4352	.4095	.2470	.1943
.105	.4534	.4257	.2247	.1822
.110	.3968	.4416	.2085	.2004
.115	.4615	.4571	.2611	.2207
.120	.4798	.4723	.2733	.2368
.125	.4818	.4871	.2976	.2308
.130	.5081	.5016	.3077	.2935
.135	.5526	.5157	.3300	.2551
.140	.5162	.5296	.3462	.3117
.145	.5122	.5431	.3563	.3421
.150	.5304	.5563	.3826	.3219

APPENDIX B

PRE-CODER

```
      SUBROUTINE PRCODE(ALPH,AMES,NMES,NCODE)
      DIMENSION ALPH(32),AMES(500),NCODE(2500),M(32,5)
      DO 15 J=1,2500
   15 NCODE(J)=0
      DO 25 N=1,32
      J=N-1
      DO 35 K=1,5
      M(N,-K+6)=MOD(J,2)
      J=J-M(N,-K+6)
   35 J=J/2
   25 CONTINUE
      DO 10 N=1,NMES
      DO 45 J=1,32
      IF(AMES(N).NE.ALPH(J)) GO TO 45
      DO 55 K=1,5
      NCODE(5*(N-1)+K)=M(J,K)
   55 CONTINUE
   45 CONTINUE
   10 CONTINUE
      RETURN
      END
```

The 32 symbols ABCDEFGHIJKLMNOPQRSTUVWXYZ blank .,:() were selected as the English text alphabet. Each is associated with one of the 32 binary 5-tuples representing the integers 0 to 31 in binary form. Any English message of length L in this alphabet is converted by the pre-coder into a continuous string of 5-tuples of zeros and ones of length 5L under this association.

Input:

ALPH is a vector of dimension 32 whose elements are the 32 English symbols.

AMES is a vector of dimension 500 which contains the English message.

NMES is the length of the English message.

Output:

NCODE is a vector of dimension 2500. It is the string of binary 5-tuples into which the English message is converted.

APPENDIX C

ENCODER

```
      SUBROUTINE ENCODE(IFCN,N,K,NMES,NCODE,NCIPH)
      DIMENSION IFCN(20),NCODE(2500),NCIPH(6000)
      L=(5*NMES)/K
      IF((L*K).NE.(5*NMES)) L=L+1
      DO 12 I=1,L
      K1=(I-1)*K
      N1=(I-1)*N
      DO 24 J=1,K
   24 NCIPH(N1+J)=NCODE(K1+J)
      NK=N-K
      N1=N1+K
      DO 12 J=1,NK
      K1=N1+J-K-1
      NSUM=0
      DO 36 L1=1,K
   36 NSUM=NSUM+(IFCN(L1)*NCIPH(K1+L1))
      NCIPH(N1+J)=MOD(NSUM,2)
   12 CONTINUE
      RETURN
      END
```

A k^{th} order linear recursion modulo 2 is selected in advance and is input to the encoder. The binary output of the pre-coder is segmented into k-tuples. Each k-tuple is used as an initial value in the linear recursion to generate an n-tuple (n selected in advance). The output of the encoder is the continuous string of n-tuples.

Input:

IFCN is a vector of dimension 20 whose first k elements are the coefficients $\alpha_k, \alpha_{k-1}, \cdots, \alpha_1$ of the linear recursion.

N and K are the integers associated with the particular linear recursive group code.

NMES is the length of the English message.

NCODE is a vector of dimension 2500 whose first 5*NMES entries form the binary string which is to be encoded.

Output:

NCIPH is a vector of dimension 6000. It is the continuous string of n-tuples generated from the k-tuples by the recursion.

APPENDIX D

CHANNEL

```
      SUBROUTINE CHANNL(II,NCIPH,NOUT,NN,P)
      DIMENSION NCIPH(6000),NOUT(6000)
      DO 70 I=1,NN
      IERR=0
      CALL RANDU(II,J,X)
      II=J
      IF(X.LE.P) IERR=1
      NOUT(I)=MOD(NCIPH(I)+IERR,2)
   70 CONTINUE
      RETURN
      END
```

The channel is a subroutine simulation of a binary symmetric channel. Input to the channel consists of a binary string and the probability P of an error (per binary digit). A uniformly-distributed random number generator is used to decide (per digit) whether or not to introduce an error. The output of the channel is this perturbed binary string.

Input:

II is any positive odd integer used as input to RANDU.

NCIPH is a vector of dimension 6000 whose first NN positions contain the binary string to be transmitted (described in Appendix C).

P is the probability of error per binary digit.

NN = L * N where L is the first integer greater than or equal to 5*NMES/K and N is as previously defined.

Output:

NOUT is a vector of dimension 6000. It is the binary string resulting from NCIPH after errors were introduced during transmission.

APPENDIX E

SYNDROME

```
      SUBROUTINE SYNDRM(IFCN,N,K,NDROME)
      DIMENSION IFCN(20),NDROME(1000),IP(20,20),NTMP(20),ISN(20)
      NK=N-K
      DO 1 I=1,K
      DO 2 J=1,K
    2 NTMP(J)=0
      NTMP(I)=1
      DO 3 J=1,NK
      NSUM=0
      DO 4 L=1,K
    4 NSUM=NSUM+IFCN(L)*NTMP(L)
      IP(I,J)=MOD(NSUM,2)
      DO 5 L=2,K
    5 NTMP(L-1)=NTMP(L)
      NTMP(K)=IP(I,J)
    3 CONTINUE
    1 CONTINUE
      IWOK=2**K
      NSYN=(2**NK)-1
      DO 6 IST=1,NSYN
      J=IST
      DO 7 L=1,NK
      ISN(L)=MOD(J,2)
      J=(J-ISN(L))/2
    7 CONTINUE
      NCONT1=N
      DO 8 NVECT=1,IWOK
      J=NVECT
      NWGT=0
      DO 9 L=1,K
      NTMP(L)=MOD(J,2)
      J=(J-NTMP(L))/2
      IF(NTMP(L).EQ.1) NWGT=NWGT+1
    9 CONTINUE
      DO 10 J=1,NK
      NSUM=0
      DO 11 L=1,K
   11 NSUM=NSUM+NTMP(L)*IP(L,J)
      NSUM=NSUM+ISN(J)
      NTMP(K+J)=MOD(NSUM,2)
      IF(NTMP(K+J).EQ.1) NWGT=NWGT+1
   10 CONTINUE
      IF(NWGT.GE.NCONT1) GO TO 12
      NCONT1=NWGT
      NCONT2=0
```

```
      J=1
      DO 13 L=1,N
      NCONT2=NCONT2+J*NTMP(L)
      J=2*J
   13 CONTINUE
   12 CONTINUE
    8 CONTINUE
      NDROME(IST)=NCONT2
    6 CONTINUE
      RETURN
      END
```

This subroutine calculates a minimum-weight n-tuple (coset leader) in each coset of the code and stores this coset leader with the syndrome of that coset.

Input:

IFCN, N, and K are described in Appendix C.

Output:

NDROME is a vector of dimension 1000 whose first $2^{N-K} - 1$ elements represent the following: the integer in the i^{th} entry is the decimal number whose binary expansion is the coset leader having syndrome given by the binary expansion of i.

APPENDIX F

DECODER

```
      SUBROUTINE DECODE(IFCN,N,K,NDROME,NMES,NOUT,NPLAIN)
      DIMENSION IFCN(20),NDROME(1000),NOUT(6000),NPLAIN(2500),
     *IP(20,20),NTMP(20)
      NK=N-K
      DO 1 I=1,K
      DO 2 J=1,K
    2 NTMP(J)=0
      NTMP(I)=1
      DO 3 J=1,NK
      NSUM=0
      DO 4 L=1,K
    4 NSUM=NSUM+IFCN(L)*NTMP(L)
      IP(I,J)=MOD(NSUM,2)
      DO 5 L=2,K
    5 NTMP(L-1)=NTMP(L)
      NTMP(K)=IP(I,J)
    3 CONTINUE
    1 CONTINUE
      L=(5*NMES)/K
      IF((L*K).NE.(5*NMES)) L=L+1
      DO 6 IWD=1,L
      LN=(IWD-1)*N
      LK=(IWD-1)*K
      JJ=1
      IST=0
      DO 7 I=1,NK
      NSUM=0
      DO 8 J=1,K
    8 NSUM=NSUM+NOUT(LN+J)*IP(J,I)
      NSUM=NOUT(LN+K+I)+NSUM
      NSUM=MOD(NSUM,2)
      IST=IST+NSUM*JJ
      JJ=2*JJ
    7 CONTINUE
      IAD=0
      IF(IST.EQ.0) GO TO 9
      IAD=NDROME(IST)
    9 CONTINUE
      DO 10 J=1,K
      JJ=MOD(IAD,2)
      NPLAIN(LK+J)=MOD(NOUT(LN+J)+JJ,2)
      IAD=(IAD-JJ)/2
   10 CONTINUE
    6 CONTINUE
      RETURN
      END
```

The pairing of coset leader with syndrome performed in the syndrome subroutine is provided as input to the decoder. The decoder segments the output of the channel into n-tuples, calculates the syndrome of each n-tuple, adds the associated coset leader to the original n-tuple, recovers the first k digits of this new n-tuple, and then outputs the continuous string of these k-tuples.

Input:

IFCN, N, K, and NMES are described in Appendix C.

NOUT is described in Appendix D.

NDROME is described in Appendix E.

Output:

NPLAIN is a vector of dimension 2500 whose first 5*NMES elements are the k-tuples of information that are recovered using maximum likelihood decoding (hopefully without error) by the decoder.

APPENDIX G

POST-CODER

```
      SUBROUTINE POCODE(NPLAIN,NMES,ALPH,ACIPH)
      DIMENSION NPLAIN(2500),ALPH(32),ACIPH(500)
      DO 40 I=1,NMES
      IJ=0
      DO 50 J=1,5
      IJ=IJ+NPLAIN(5*(I-1)+J)*(2**(-J+5))
   50 CONTINUE
      IJ=IJ+1
      ACIPH(I)=ALPH(IJ)
   40 CONTINUE
      RETURN
      END
```

The post-coder receives a binary string, segments this string into 5-tuples and converts each 5-tuple into an alphabet symbol by the same association used in the pre-coder.

Input:

ALPH and NMES are described in Appendix B

NPLAIN is described in Appendix F

Output:

ACIPH is a vector of dimension 500 whose first NMES elements contain the decoded message which may not be the same as the original message transmitted.